Quantum Circuits: From Classical to Quantum Computation

Quantum Computing Course

Lecture Duration: 90 minutes

Contents

1 From Physical Operations to Computation

1.1 Positioning Quantum Circuits in the Course

Welcome to our lecture on quantum circuits. In previous lectures, we have systematically built the foundation:

- **Dirac's notation** provided the language for quantum states
- **Quantum mechanics basics** established superposition and measurement
- **Qubits** defined our fundamental unit of quantum information
- **Quantum gates** gave us the operations: X, Y, Z, H, CNOT, etc.
- **Unitary and Hermitian matrices** provided the mathematical framework

Today, we reach a crucial synthesis: **the quantum circuit model**. This is where isolated concepts become a coherent computational framework. The circuit model serves as the "programming language" of quantum computing—the standard way to describe quantum algorithms, analyze their complexity, and understand their behavior.

> **Key Insight**
>
> A quantum circuit is more than just a collection of gates; it's a structured representation of quantum information flow through time, subject to the fundamental constraints of quantum mechanics.

1.2 Why a Circuit Model is Needed: Abstraction and Universality

Why do we need this particular abstraction? Several reasons:

1. **Standardization:** Just as classical algorithms are described using flowcharts or pseudocode, quantum algorithms need a standard representation.
2. **Universality:** The circuit model is universal for quantum computation. Any quantum computation can be expressed as a quantum circuit (with reasonable assumptions about resources).
3. **Visualization:** Circuit diagrams provide an intuitive visual representation of complex quantum operations.
4. **Analysis:** Circuits allow us to analyze computational resources: qubit count (width), time steps (depth), and gate count.
5. **Implementation Planning:** Real quantum computers are programmed using circuit descriptions.

The quantum circuit model is to quantum computing what Boolean circuits are to classical computing: a fundamental model of computation that abstracts away physical details while preserving computational essence.

1.3 Scope, Assumptions, and Learning Outcomes

1.3.1 Scope of This Lecture

We will focus on:

- The **syntax and semantics** of quantum circuit diagrams
- How to **read, interpret, and construct** simple quantum circuits
- **Fundamental differences** from classical Boolean circuits
- **Practical conventions** used in quantum computing literature

We will **not** cover:

- Specific quantum algorithms in detail (these come later)
- Physical implementations of circuits
- Error correction and fault tolerance

1.3.2 Key Assumptions

1. We work with **idealized** qubits and gates (no decoherence, perfect operations)
2. All gates are **unitary** (except measurement, which is special)
3. We have access to a **universal gate set** (we'll define this more precisely in future lectures)

1.3.3 Learning Outcomes

By the end of this lecture, you should be able to:

1. Read and interpret quantum circuit diagrams
2. Construct simple quantum circuits for basic tasks
3. Analyze the state evolution through a circuit
4. Understand and explain the fundamental differences between quantum and classical circuits
5. Apply standard conventions when drawing quantum circuits

2 Classical Circuits as a Computational Model

2.1 Circuit-Based Computation: An Overview

Before diving into quantum circuits, let's establish our classical baseline. Classical Boolean circuits are a standard model of computation where:

Definition 1 (Classical Boolean Circuit)**.** *A classical Boolean circuit is a directed acyclic graph where:*

- ***Nodes*** *are logic gates (AND, OR, NOT, etc.)*
- ***Edges*** *(wires) carry bits (0 or 1)*
- *Computation flows from inputs (left) to outputs (right)*
- *Each gate computes a Boolean function (typically* $\{0,1\}^k \to \{0,1\}$*)*

Representation:

$$\begin{array}{c}
x \longrightarrow \boxed{\text{Circuit}} \longrightarrow f_1(x,y,z) \\
y \longrightarrow \boxed{\phantom{\text{Circuit}}} \longrightarrow f_2(x,y,z) \\
z \longrightarrow \boxed{\phantom{\text{Circuit}}} \longrightarrow f_3(x,y,z)
\end{array}$$

Key Properties:

- **Deterministic:** Same input always produces same output
- **Compositional:** Complex circuits built from simple gates
- **Finite:** Fixed number of gates and wires

2.2 Irreversibility and Information Loss

Most classical gates are **irreversible**: you cannot determine the inputs from the outputs.

Example 1 (AND Gate Irreversibility)**.** *Consider an AND gate:* $f(x,y) = x \wedge y$

- *Output 0: Could be* $(0,0)$, $(0,1)$, *or* $(1,0)$
- *Output 1: Must be* $(1,1)$

Given output 0, we cannot uniquely determine the inputs. Information is lost.

Information-Theoretic Perspective

Irreversible gates dissipate energy (Landauer's principle) and lose information. For an n-input, m-output gate with $n > m$, the mapping $\{0,1\}^n \to \{0,1\}^m$ is many-to-one, so information about the input is irrevocably lost.

2.3 Reversible Classical Circuits

Interestingly, not all classical computation needs to be irreversible. **Reversible classical circuits** use only reversible gates.

2.3.1 Classical Controlled-NOT and Toffoli Gates

Definition 2 (Classical CNOT Gate)**.** *The classical Controlled-NOT (CNOT) gate has 2 inputs* (c, t) *and 2 outputs* $(c, t \oplus c)$*:*

- *If control* $c = 0$*: target* t *passes through unchanged*
- *If control* $c = 1$*: target* t *is flipped (NOT applied)*

c (control)	t (target)	c'	t'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Definition 3 (Toffoli Gate (CCNOT))**.** *The Toffoli gate has 3 inputs* (c_1, c_2, t) *and outputs* $(c_1, c_2, t \oplus (c_1 \wedge c_2))$*. It flips the target only if both controls are 1.*

Key Fact

The Toffoli gate is **universal** for reversible classical computation: any reversible Boolean function can be implemented using only Toffoli gates.

2.3.2 Fan-out and Information Copying

A crucial feature of classical circuits is **fan-out**: the ability to copy a bit onto multiple wires.

$$\begin{array}{c} x \longrightarrow \boxed{\text{COPY}} \longrightarrow x \\ 0 \longrightarrow \boxed{\phantom{\text{COPY}}} \longrightarrow x \end{array}$$

This classical CNOT with target initialized to 0 performs copying: $(x, 0) \to (x, x)$. This is fundamental to classical computation and taken for granted.

2.4 Limitations of the Classical Circuit Paradigm

2.4.1 Circuit Universality (Brief Mention)

In classical computing, we have the concept of **universal gate sets**: collections of gates that can compute any Boolean function. For example:

- {NAND} is universal for classical computation

- {Toffoli} is universal for reversible classical computation

We'll see that quantum computing has its own universality concepts, but they're more constrained due to unitarity.

Transition to Quantum

Classical circuits give us a familiar foundation, but quantum circuits operate under fundamentally different rules. The constraints are stricter (unitarity), but the capabilities are vastly expanded (superposition, entanglement).

3 The Quantum Circuit Model

3.1 Quantum Circuits as Structured Unitary Evolution

Definition 4 (Quantum Circuit)**.** *A quantum circuit is a sequence of quantum gates and measurements applied to a collection of qubits, represented as a diagram where:*

- ***Horizontal lines*** *(wires) represent qubits over time*
- ***Boxes*** *represent quantum gates (unitary operations)*
- ***Time flows left to right***
- *Quantum gates (excluding measurements) implement unitary transformations:* $U = U_k \cdots U_2 U_1$

Example Circuit Representation:

$$\begin{array}{c}
|\psi_0\rangle \longrightarrow \boxed{U_1} \longrightarrow \boxed{U_2} \longrightarrow \boxed{U_3} \longrightarrow M \longrightarrow \text{Bit} \\
|0\rangle \longrightarrow \boxed{V_1} \longrightarrow \boxed{V_2} \longrightarrow \qquad \longrightarrow M \longrightarrow \text{Bit} \\
|0\rangle \longrightarrow \qquad \longrightarrow \qquad \longrightarrow \boxed{W_1} \longrightarrow \boxed{W_2} \longrightarrow |\phi\rangle
\end{array}$$

This circuit shows:

- Three qubits with different initial states
- Single-qubit gates $(U_1, U_2, V_1, W_1, W_2)$
- Two-qubit gates (U_3, V_2)
- Measurements on the first two qubits (producing classical bits)
- One qubit remaining in quantum state at the end

3.2 Why Must Quantum Gates Be Unitary?

This is a fundamental question with answers at multiple levels:

3.2.1 Physical Reason: Closed System Evolution

Quantum mechanics tells us that the evolution of a **closed quantum system** is described by the Schrödinger equation:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

where $H(t)$ is the Hamiltonian (Hermitian operator). The solution is:

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle$$

where $U(t)$ is a **unitary operator**: $U^\dagger U = UU^\dagger = I$.

Physical Interpretation

Unitary evolution preserves the total probability (norm = 1) because closed quantum systems don't lose or gain probability—they just redistribute it. Unitary evolution describes isolated systems or subsystems treated as closed.

3.2.2 Mathematical Reason: Preservation of Norm and Inner Products

Unitarity ensures two crucial mathematical properties:

1. **Norm preservation:** $\|U|\psi\rangle\| = \||\psi\rangle\|$
   - This means probabilities sum to 1: $\sum_i |\alpha_i|^2 = 1$
   - Essential for probability interpretation of quantum mechanics
2. **Inner product preservation:** $\langle U\phi|U\psi\rangle = \langle\phi|\psi\rangle$
   - Relative phases and overlaps between states are preserved
   - This enables quantum interference, a key computational resource

3.2.3 Computational Reason: Reversibility and Information Preservation

Unitarity implies **reversibility**: for every unitary gate U, there exists an inverse gate $U^\dagger$ such that:

$$U^\dagger U = UU^\dagger = I$$

This means (up to measurement) quantum computations are reversible—you can theoretically "run the circuit backward."

Computational Implication

Unlike classical gates that can erase information (like AND), quantum gates must preserve all information about the input state (except what's deliberately discarded through measurement). This is both a constraint and a feature.

Example 2 (Non-unitary Operations Are Not Allowed)**.** *Consider an operation that "resets" a qubit to* $|0\rangle$*:*

$$E : \alpha|0\rangle + \beta|1\rangle \to |0\rangle$$

*This is **not unitary** because:*

- *Different inputs (*$|0\rangle$ *and* $|1\rangle$*) map to the same output*
- *The operation is not invertible (cannot recover* α, β *from* $|0\rangle$*)*
- *Norm is not preserved unless* $|\alpha|^2 = 1$

Such operations require interaction with an environment (open system) or measurement.

3.3 Circuit Diagrams: Syntax and Meaning

Let's formalize the syntax of quantum circuit diagrams.

3.3.1 Quantum Wires and Time Flow

$$|\psi\rangle \longrightarrow \text{(time flow)} \longrightarrow |\psi'\rangle$$

- Each horizontal line represents **one qubit**
- The line traces the qubit's existence from preparation to measurement/discard
- **Time increases from left to right**
- Wires are not physical wires but worldlines of quantum systems

3.3.2 Gate Notation and Placement

$$\begin{array}{c} |q_0\rangle \longrightarrow \boxed{H} \longrightarrow \bullet \longrightarrow \boxed{R_z(\theta)} \longrightarrow \\ |q_1\rangle \longrightarrow \qquad \longrightarrow \oplus \longrightarrow \qquad\qquad \longrightarrow \textcircled{M} \longrightarrow \text{Bit} \end{array}$$

Gate placement rules:

1. **Single-qubit gates** sit on a single wire
2. **Multi-qubit gates** span multiple wires
3. Gates on the same wire are applied sequentially
4. Gates on different wires at the same horizontal position are applied in parallel

Circuit Diagram Legend:

$\bullet$ = Control qubit $\quad \oplus$ = Target qubit
Box spanning multiple wires = Multi-qubit gate
$\textcircled{M}$ = Measurement in computational basis

3.3.3 Measurement Symbols and Classical Outputs

$$|\psi\rangle \longrightarrow \textcircled{M} \longrightarrow \text{(classical bit)}$$

- The meter symbol $\textcircled{M}$ represents measurement in computational basis
- Measurement produces a classical bit (0 or 1)
- Double lines represent classical bits/wires
- Measurement is **not unitary**—it's probabilistic and irreversible

3.4 Time Ordering and Information Flow

Quantum circuits have precise time ordering:

Principle 1 (Time Ordering in Quantum Circuits)**.** *Operations are applied in the order: left $\rightarrow$ right, with all gates at the same horizontal position applied simultaneously (in parallel).*

Mathematically, if gates $U_1, U_2, \ldots, U_n$ are applied sequentially, the overall unitary is:

$$U = U_n \cdots U_2 U_1$$

Note the reverse order: the rightmost gate in the formula is the leftmost in the circuit!

Example 3 (Time Ordering Example)**.**

$$|\psi\rangle \longrightarrow \boxed{A} \longrightarrow \boxed{B} \longrightarrow \boxed{C} \longrightarrow CBA|\psi\rangle$$

The circuit applies A, then B, then C, so the final state is $CBA|\psi\rangle$.

Parallel Operations

When gates are on different qubits at the same horizontal position, they represent tensor products:

$$\begin{array}{c} \longrightarrow \boxed{U} \longrightarrow \\ \longrightarrow \boxed{V} \longrightarrow \end{array} \quad \text{represents} \quad U \otimes V$$

3.5 Measurement as the Computational Interface

Measurement plays a special role: it's the interface between quantum and classical information.

- **Before measurement:** Pure quantum evolution (unitary)
- **During measurement:** Probabilistic collapse (or state update) to computational basis
- **After measurement:** The quantum state is projected and treated as classical information that we can read, copy, and process classically

Measurement Convention

Unless otherwise stated, all measurements are assumed to be in the **computational basis** $\{|0\rangle, |1\rangle\}$.

4 Interpreting Quantum Circuits

4.1 State Evolution Through a Circuit

4.1.1 Algebraic Interpretation

To interpret a circuit algebraically:

1. Start with initial state: $|\psi_0\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$
2. Apply gates sequentially from left to right
3. For parallel gates, take tensor products
4. For gates spanning multiple qubits, apply the appropriate multi-qubit unitary

Example 4 (Algebraic Interpretation)**.** *Consider:*

$$\begin{array}{c} |0\rangle \longrightarrow \boxed{H} \longrightarrow \bullet \longrightarrow \\ |0\rangle \longrightarrow \qquad \longrightarrow \oplus \longrightarrow \end{array}$$

Step-by-step:

$$\begin{aligned} \textit{Initial:} \quad & |00\rangle \\ \textit{After H:} \quad & H \otimes I\,|00\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\ \textit{After CNOT:} \quad & CNOT\left[\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)\right] = \frac{1}{\sqrt{2}}(CNOT|00\rangle + CNOT|10\rangle) \\ = & \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \end{aligned}$$

4.1.2 Diagrammatic Interpretation

Circuit diagrams also allow "intuitive" reading without full algebra:

$$\begin{array}{c} |0\rangle \longrightarrow \boxed{H} \longrightarrow \bullet \longrightarrow M \longrightarrow m_1 \\ |0\rangle \longrightarrow \qquad \longrightarrow \oplus \longrightarrow M \longrightarrow m_2 \end{array}$$

We can read this as:

- "Put first qubit in superposition"
- "Entangle the two qubits"
- "Measure both qubits"
- "The measurements will be correlated"

4.2 Implicit Tensor Products and Identity Operations

An important convention: when a gate is applied to some qubits, the identity operation I is implicitly applied to all other qubits.

Example 5 (Implicit Identities)**.**

$$\begin{array}{l} |\psi_1\rangle \longrightarrow \boxed{U} \longrightarrow \\ |\psi_2\rangle \longrightarrow \qquad \longrightarrow \\ |\psi_3\rangle \longrightarrow \qquad \longrightarrow \end{array} \quad = \quad U \otimes I \otimes I$$

The U gate acts only on the first qubit; qubits 2 and 3 experience identity operations.

4.3 Parallelism and Circuit Layers

We can organize circuits into **layers**—sets of gates that can be applied in parallel.

Definition 5 (Circuit Layer)**.** *A layer is a set of gates at the same horizontal position in a circuit diagram. All gates in a layer:*

- *Are applied simultaneously*
- *Act on disjoint sets of qubits*
- *Can be implemented in parallel in ideal hardware*

$$\begin{array}{c} \longrightarrow \boxed{H} \longrightarrow \bullet \longrightarrow \boxed{X} \longrightarrow \bullet \longrightarrow \\ \longrightarrow \boxed{H} \longrightarrow \oplus \longrightarrow \boxed{Y} \longrightarrow \oplus \longrightarrow \\ \longrightarrow \boxed{Z} \longrightarrow \qquad \longrightarrow \bullet \longrightarrow \boxed{H} \longrightarrow \end{array}$$

This circuit has 5 layers. Layer 2 (CNOT) and Layer 4 (CNOT) are two-qubit gates; other layers have single-qubit gates that can be parallelized.

4.4 Measurement Outcomes and Classical Data

When we measure, we get classical data that can:

1. Be read out as the computation result
2. Control subsequent classical processing
3. Sometimes control subsequent quantum operations (classically controlled gates)

$$\begin{array}{l} |\psi\rangle \longrightarrow M \longrightarrow \text{(controls)} \\ |0\rangle \longrightarrow \qquad \longrightarrow \boxed{X_{\text{conditional}}} \end{array}$$

Here, the measurement outcome controls whether an X gate is applied to another qubit. This is a **classically controlled quantum gate**.

Important Distinction

Quantum control (like CNOT): Control is quantum superposition
Classical control (like shown above): Control is a definite classical bit

5 Representative Circuit Constructions

5.1 Example 1: Entangling Circuit Patterns

5.1.1 Two-Qubit Entanglement Generation

The canonical two-qubit entangling operation uses Hadamard followed by CNOT:

$$\begin{array}{ccccccc}
|0\rangle & \longrightarrow & \boxed{H} & \longrightarrow & \bullet & \longrightarrow \\
|0\rangle & \longrightarrow & & \longrightarrow & \oplus & \longrightarrow
\end{array}$$

Let's trace the evolution:

$$\begin{aligned}
|00\rangle &\xrightarrow{H\otimes I} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\
&\xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\end{aligned}$$

This creates a **maximally entangled state**: measuring one qubit immediately determines the other, regardless of distance (in principle).

5.1.2 Multi-Qubit Correlation Circuits

We can extend this pattern to create complex correlations:

$$\begin{array}{ccccccccccc}
|0\rangle & \longrightarrow & \boxed{H} & \longrightarrow & \bullet & \longrightarrow & & \longrightarrow & \bullet & \longrightarrow \\
|0\rangle & \longrightarrow & & \longrightarrow & \oplus & \longrightarrow & \bullet & \longrightarrow & & \longrightarrow \\
|0\rangle & \longrightarrow & & \longrightarrow & & \longrightarrow & \oplus & \longrightarrow & \oplus & \longrightarrow
\end{array}$$

This circuit creates a three-qubit correlated state. Notice the pattern:

- Start with all qubits in $|0\rangle$
- Apply H to first qubit to create superposition
- Use CNOTs to ”spread” the superposition, creating correlations
- The specific pattern of CNOTs determines the correlation structure

5.2 Example 2: State Permutation Circuits

5.2.1 Quantum SWAP Circuit

Notation: CNOT_{ij} denotes control on qubit i, target on qubit j.

A fundamental operation is swapping the states of two qubits. The quantum SWAP can be implemented with three CNOTs:

$$\begin{array}{ccccccccc}
|\psi\rangle & \longrightarrow & \bullet & \longrightarrow & \oplus & \longrightarrow & \bullet & \longrightarrow & |\phi\rangle \\
|\phi\rangle & \longrightarrow & \oplus & \longrightarrow & \bullet & \longrightarrow & \oplus & \longrightarrow & |\psi\rangle
\end{array}$$

Let's verify this works for computational basis states:

- Input $|00\rangle$: All CNOTs do nothing → output $|00\rangle$
- Input $|01\rangle$:

$$
\begin{aligned}
|01\rangle &\xrightarrow{\text{CNOT}_{12}} |01\rangle \quad (\text{control} = 0, \text{target unchanged}) \\
&\xrightarrow{\text{CNOT}_{21}} |11\rangle \\
&\xrightarrow{\text{CNOT}_{12}} |10\rangle
\end{aligned}
$$

  Swapped!

- Similarly for $|10\rangle$ and $|11\rangle$

By linearity, it works for any superposition state.

5.2.2 Controlled Swap and Exchange Operations

We can also make controlled versions of swaps:

$$
\begin{array}{c}
|c\rangle \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow |c\rangle \\
|\psi\rangle \longrightarrow \oplus \longrightarrow \bullet \longrightarrow \oplus \longrightarrow |\psi\rangle \text{ or } |\phi\rangle \\
|\phi\rangle \longrightarrow \quad \longrightarrow \oplus \longrightarrow \quad \longrightarrow |\phi\rangle \text{ or } |\psi\rangle
\end{array}
$$

This is a **controlled-SWAP** (Fredkin gate): swaps the bottom two qubits only if the top (control) qubit is $|1\rangle$.

5.3 Example 3: Multi-Stage Processing Networks

5.3.1 Sequential Gate Applications

Circuits often involve sequences of different operations:

$$|0\rangle \longrightarrow \boxed{H} \longrightarrow \boxed{S} \longrightarrow \boxed{T} \longrightarrow \boxed{H} \longrightarrow \boxed{X} \longrightarrow \boxed{Z} \longrightarrow$$

This shows a single qubit undergoing multiple rotations and flips. The overall operation is:

$$Z \cdot X \cdot H \cdot T \cdot S \cdot H$$

(remember: right-to-left in formula = left-to-right in circuit)

5.3.2 Parallel Gate Operations

Modern quantum algorithms exploit parallelism:

$$
\begin{array}{c}
|0\rangle \longrightarrow \boxed{H} \longrightarrow \boxed{\text{U (4-qubit)}} \longrightarrow \textcircled{M} \longrightarrow m_1 \\
|0\rangle \longrightarrow \boxed{H} \longrightarrow \qquad\qquad\quad \longrightarrow \textcircled{M} \longrightarrow m_2 \\
|0\rangle \longrightarrow \boxed{H} \longrightarrow \qquad\qquad\quad \longrightarrow \textcircled{M} \longrightarrow m_3 \\
|0\rangle \longrightarrow \boxed{H} \longrightarrow \qquad\qquad\quad \longrightarrow \textcircled{M} \longrightarrow m_4
\end{array}
$$

Here:

- All four qubits are put in superposition in parallel (Layer 1)
- A large 4-qubit unitary U is applied (Layer 2)
- All qubits are measured in parallel (Layer 3)

This pattern appears in many quantum algorithms: parallel preparation $\rightarrow$ joint processing $\rightarrow$ parallel measurement.

6 Quantum vs. Classical Circuits: A Conceptual Comparison

6.1 Correlation, Interference, and Information Encoding

6.1.1 Superposition vs. Definite States

Classical Wires	**Quantum Wires**
Carry definite values: 0 or 1	Carry superpositions: $\alpha\|0\rangle + \beta\|1\rangle$
State is always a basis state	State can be any unit vector in Hilbert space
No phase information	Phase ($\arg(\alpha/\beta)$) matters critically

The superposition principle means a single quantum wire encodes a continuum of states, while a classical wire encodes just 1 bit.

6.1.2 Entanglement vs. Classical Correlation

Classical Correlation	**Quantum Entanglement**
Described by joint probability distributions	Described by non-separable state vectors
Can always be explained by shared randomness	Exhibits non-local correlations (Bell inequalities)
Correlations obey classical bounds	Can violate classical correlation bounds
All states are separable: $p(x, y)$	Can be non-separable: $\|\psi\rangle \neq \|\psi_1\rangle \otimes \|\psi_2\rangle$

Entanglement enables quantum circuits to process information in ways impossible classically, like in quantum teleportation and superdense coding.

6.2 Reversibility as a Fundamental Constraint

6.2.1 Unitarity and Information Preservation

Classical Gates	**Quantum Gates**
Most are irreversible (AND, OR, NAND)	All (except measurement) are unitary (reversible)
Information can be erased	Information preserved (up to measurement)
Many-to-one mappings allowed	Must be one-to-one (bijective)
Thermodynamically dissipative	Logically reversible (no fundamental Landauer cost)

The reversibility of quantum gates means quantum circuits (before measurement) are invertible. Given the output quantum state, you could apply the inverse circuit to recover the input.

Computational Consequence

Classical computers need ”garbage collection” to manage information erasure. Quantum computers, in principle, don’t—but they need to manage reversibility, which introduces its own complexities (like needing ancilla qubits).

6.3 Determinism vs. Probabilistic Outcomes

6.3.1 Measurement Disturbance

Classical Measurement	**Quantum Measurement**
Non-invasive: reading a bit doesn’t change it	Disturbing: measurement collapses (updates) the state
Deterministic: same value every time	Probabilistic: outcomes follow Born rule
Can copy before/after measurement	Cannot copy arbitrary states (No-Cloning)
Can measure partially (some bits)	Measurement updates the global state of an entangled system

Most quantum algorithms need to be run multiple times to gather statistics (though some deterministic algorithms exist).

6.4 Restrictions Unique to Quantum Circuits

6.4.1 No-Cloning Theorem (Statement and Implications)

Principle 2 (No-Cloning Theorem)**.** *There is no unitary operation U and fixed state $|s\rangle$ such that for all $|\psi\rangle$:*

$$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle$$

Implications for circuit design:

- **No fan-out:** Cannot copy a quantum state onto multiple wires
- **No backup:** Cannot make copies for error checking (need quantum error correction instead)
- **Measurement limitation:** Cannot measure without disturbing
- **Algorithm design:** Quantum algorithms must work without copying intermediate states

6.4.2 No-Deleting and No-Broadcasting

Related restrictions:

- **No-Deleting:** Cannot take two copies and delete one to get back one

- **No-Broadcasting:** Cannot copy correlations (weaker than no-cloning but still restrictive)

Fundamental Quantum Constraints

Both cloning and deleting quantum information are impossible **unitarily**. Classical circuits: Can copy but cannot delete reversibly
Quantum circuits: Cannot copy or delete information unitarily; irreversible operations require measurement or environment.

7 Complexity-Oriented Circuit Properties

7.1 Circuit Width and Auxiliary Resources

7.1.1 Qubit Count and Ancilla Qubits

Definition 6 (Circuit Width)**.** *The number of qubits required to implement a circuit. This includes:*

- ***Input qubits:*** *Carrying the problem input*
- ***Output qubits:*** *Holding the final answer (often same as input)*
- ***Ancilla qubits:*** *Auxiliary qubits needed for computation*

Ancilla qubits are often initialized to $|0\rangle$ and returned to $|0\rangle$ at the end (to avoid leaving garbage that would prevent reversibility).

$$\begin{array}{c} |x\rangle \longrightarrow \boxed{U} \longrightarrow |f(x)\rangle \\ |0\rangle \longrightarrow \qquad \longrightarrow |0\rangle \\ |0\rangle \longrightarrow \qquad \longrightarrow |0\rangle \end{array}$$

Here, two ancilla qubits are used during computation but returned to $|0\rangle$.

7.1.2 Circuit Depth and Time Complexity

Definition 7 (Circuit Depth)**.** *The number of layers in a circuit—the length of the critical path. This represents the minimum number of time steps needed (with perfect parallelism).*

$$\begin{array}{c} \longrightarrow \boxed{H} \longrightarrow \bullet \longrightarrow \boxed{X} \longrightarrow \bullet \longrightarrow \\ \longrightarrow \boxed{H} \longrightarrow \oplus \longrightarrow \boxed{Y} \longrightarrow \oplus \longrightarrow \\ \longrightarrow \boxed{Z} \longrightarrow \qquad \longrightarrow \bullet \longrightarrow \boxed{H} \longrightarrow \end{array}$$

Depth analysis:

- Layer 1: H, H, Z (all in parallel) $\rightarrow$ 1 time step
- Layer 2: CNOT (between qubits 1-2) $\rightarrow$ 1 time step
- Layer 3: X, Y, CNOT-control (q3 controls q2) $\rightarrow$ 1 time step
- Layer 4: CNOT (between qubits 1-2) $\rightarrow$ 1 time step
- Layer 5: H $\rightarrow$ 1 time step

Total depth = 5 time steps.

7.2 Resource Counting: Gates and Qubits

When analyzing quantum algorithms, we count:

1. **Qubits:** Width of the circuit
2. **Gate count:** Total number of gates
3. **Depth:** Parallel time steps
4. **Specific gates:** Especially expensive gates (like T gates in fault-tolerant computing)

Example 6 (Resource Counting)**.** *A circuit with:*

- *10 qubits (width)*
- *50 single-qubit gates + 20 two-qubit gates (gate count)*
- *Depth 15 (time steps)*

Might be considered efficient or inefficient depending on the problem solved.

7.3 Idealized Circuits vs. Physical Implementations

Idealized Circuit Model	**Physical Reality**
Perfect qubits (no decoherence)	Qubits decohere over time
Perfect gates (exact unitaries)	Gates have finite error rates
Arbitrary connectivity	Limited qubit connectivity (nearest neighbor)
Unlimited parallelism	Limited parallel gate execution
Instantaneous gates	Gates take finite time

In practice, circuit design must consider:

- **Decoherence time:** Circuit depth must fit within coherence time
- **Gate fidelity:** Number of gates limited by error accumulation
- **Connectivity:** Need SWAP networks to move information
- **Gate set:** Actual hardware has native gates; others must be compiled

8 Conventions, Abstractions, and Practical Remarks

8.1 Register Structure and Qubit Ordering

8.1.1 Most Significant vs. Least Significant Convention

There are two common conventions for multi-qubit states:

1. **Most significant qubit first (top wire):**

$$|q_0 q_1 \cdots q_{n-1}\rangle \quad \text{where } q_0 \text{ is most significant}$$

   Used in many textbooks and Qiskit.

2. **Least significant qubit first (top wire):**

$$|q_{n-1} \cdots q_1 q_0\rangle \quad \text{where } q_0 \text{ is least significant}$$

   Used in some quantum algorithm papers.

> Important!
>
> Always check which convention is being used! Inconsistency causes errors in interpreting measurement outcomes and binary representations.

8.2 Classical Control and Hybrid Circuits

8.2.1 Conditional Quantum Operations

Circuits can mix quantum and classical control:

$$\begin{array}{c} |\psi\rangle \longrightarrow M \longrightarrow \text{(controls V)} \\ |\phi\rangle \longrightarrow \boxed{U} \longrightarrow M \longrightarrow \text{(controls W)} \\ |0\rangle \longrightarrow \qquad \longrightarrow \boxed{V_{\text{conditional}}} \longrightarrow \boxed{W_{\text{conditional}}} \end{array}$$

Here:

- Measurement on first qubit controls V gate
- Measurement on second qubit controls W gate
- Classical control allows adaptive circuits (measurement-based quantum computing)

8.3 Circuit Diagrams as a Programming Abstraction

Quantum circuit diagrams are the ”assembly language” of quantum computing:

- **Hardware-independent:** Same circuit can (in principle) run on different quantum hardware
- **Compilation needed:** High-level algorithms compile to circuits
- **Optimization:** Circuits can be optimized (gate merging, cancellation)
- **Verification:** Can formally verify circuit correctness

The Circuit Model's Role

The quantum circuit model sits between:

- **Above:** Quantum algorithms (abstract descriptions)
- **Below:** Physical implementations (hardware constraints)

It's the level where we can reason about computational complexity without hardware details.

9 Summary and Outlook

9.1 Key Conceptual Takeaways

1. **Quantum circuits** are the standard model for describing quantum computations, analogous to Boolean circuits for classical computation.

2. **Syntax:** Wires = qubits, boxes = gates, left-to-right = time, measurement produces classical bits.

3. **Semantics:** Quantum gates (excluding measurements) implement unitary evolution, applied sequentially and in parallel.

4. **Fundamental differences from classical:**

   - Superposition on wires (not just 0/1)
   - Entanglement between wires (non-classical correlations)
   - Unitarity (reversibility, no information erasure)
   - No-Cloning Theorem (cannot copy quantum states)
   - Probabilistic measurement (Born rule)

5. **Complexity measures:** Width (qubits), depth (time steps), gate count.

9.2 The Role of Circuits in Quantum Algorithms

In upcoming lectures, we'll see how:

- Algorithms like Deutsch-Jozsa, Grover, Shor are expressed as quantum circuits
- Circuit analysis reveals algorithm complexity
- Circuit optimization improves practical implementation
- Error correction adds structure to circuits

9.3 Preparation for Algorithmic Case Studies

To prepare for algorithm studies:

1. Practice reading and interpreting circuit diagrams
2. Work through state evolution for small circuits
3. Understand how measurement extracts classical information
4. Recognize common circuit patterns (entanglement, interference)

9.4 Suggested Reading and Exercises

9.4.1 Recommended Reading

- Nielsen & Chuang: "Quantum Computation and Quantum Information" - Chapter 4
- Kaye, Laflamme, Mosca: "An Introduction to Quantum Computing" - Chapter 4
- Preskill's Quantum Computing Notes: Chapter 3

9.4.2 Exercises for Practice

1. Draw a circuit that creates the state $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$.
2. For the SWAP circuit (three CNOTs), verify it works for input $|10\rangle$.
3. What is the depth of this circuit?

$$\begin{array}{c}\longrightarrow \boxed{H} \longrightarrow \bullet \longrightarrow \boxed{X} \longrightarrow \\ \longrightarrow \boxed{H} \longrightarrow \oplus \longrightarrow \boxed{Y} \longrightarrow\end{array}$$

4. Explain why the following cannot be a valid quantum gate:

$$G : \alpha |0\rangle + \beta |1\rangle \to \alpha |0\rangle$$

5. Design a circuit that takes $|00\rangle$ to $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$.

Final Thought

The quantum circuit model gives us a powerful language for quantum computation. While constrained by quantum mechanics (unitarity, no-cloning), it enables computational capabilities beyond classical circuits through superposition, entanglement, and interference. Mastery of this model is essential for understanding and designing quantum algorithms.